

NCC-387

November 20, 1986

AMES GRANT
7N-62-CR
J80 700
p.12

The Security of Data in Networks

Peter J. Denning

Research Institute for Advanced Computer Science
NASA Ames Research Center

RIACS Technical Report 86.26
November 20, 1986

(NASA-CR-180684) THE SECURITY DATA IN
NETWORKS (Research Inst. for Advanced
Computer Science) 12 p

N91-70134

Uncl us
00/62 0280700

RIACS

Research Institute for Advanced Computer Science

The Security of Data in Networks

Peter J. Denning

Research Institute for Advanced Computer Science
NASA Ames Research Center

RIACS Technical Report 86.26
November 20, 1986

Telescience is NASA's word for scientific research conducted via networks that permit remote control of experiments and collaboration of scientists around the world on analyzing the results. The safety of remotely controlled experiments and integrity of research rest critically on the ability of the network to authenticate senders and receivers, to protect proprietary communications, and to sign some transmissions. Mathematically sound schemes for encrypting data and distributing keys make these goals attainable.

Work reported herein was supported in part by Cooperative Agreement NCC 2-387 between the National Aeronautics and Space Administration (NASA) and the Universities Space Research Association (USRA).

This is a preprint of the column *The Science of Computing* for *American Scientist* 75, No 1, January-February, 1987.

The Security of Data in Networks

Peter J. Denning

Research Institute for Advanced Computer Science

November 20, 1986

Telescience. This term is used by NASA to refer to scientific research conducted with computers and instruments connected by networks over great distances. It includes the remote design of experiments on space platforms, the operation of those experiments, and the collaboration of scientists around the world in interpreting data and publishing results. The next best thing to being there, telescience is expected to be a common mode of research in all scientific fields by the mid 1990s.

For the safety of remotely-controlled operations and the integrity of their research, experimenters want to be certain that they are linked to their own instruments when they request connections and that no one else can connect to those instruments. They want to be certain that no one can alter the data transmitted from their instruments, or the authorized commands sent to the instruments. They want to be certain that proprietary communications with their co-workers cannot be disclosed. The first guarantee, called authentication,

certifies the identity of a principal -- person, computer, or device -- accessible on the network. The second guarantee, called integrity, certifies that a data stream actually comes from a previously authenticated source. The third guarantee, called secrecy, certifies that the content of a data stream is hidden from outside view. Data transmissions covered by these guarantees are called secure communications. Telescience requires secure communications over high-bandwidth networks -- 1 million bits per second (Mbps) or more.

Who furnishes these guarantees? The agencies that design and operate a network must provide for them in the communications protocols. All such mechanisms ultimately require that each principal can possess or obtain information that identifies any other principal. The identifying information can be embodied as a key to encipher data. The mechanisms must be capable not only of efficiently enciphering and deciphering data, but of distributing and protecting keys. In what follows, I will present a brief survey of this fascinating subject. A comprehensive treatment can be found in Dorothy Denning's book *Cryptography and Data Security*

Communication between principals can be a two-way conversation in real time, a one-way, high-rate data stream, or a one-way mail or datagram message. Some communications must be signed by attaching an unforgeable mark that will establish the sender's identity beyond reasonable doubt.

A communications path through a network may include many links, switches, computers, local networks, and internetwork gateways. In most

networks these components are vulnerable because data security was not a requirement of the original design. Each component is a potential site for an intruder to eavesdrop on a conversation, read mail, replay portions of prior messages, or alter a data transmission. Because a pair of principals wishing to communicate have no control over these many network components, they must use protocols that allow them to control the encryption devices and the keys.

Traditional cryptosystems are based on a single key K known only to A and B , the principals who wish to communicate. A message M is sent as ciphertext, denoted $[M]^K$. This scheme provides authentication as well as secrecy: if an attempt by B to decipher a message produces gibberish, B knows that A could not have been the sender.

The best known computer-based cryptosystem is the Data Encryption Standard (DES), promulgated in 1977 by the National Bureau of Standards. The DES uses a 56-bit key to encipher successive 64-bit blocks of data. Computer chips embodying the DES algorithm operate at speeds beyond 10 Mbps, which is faster than needed for most wide-area communication networks. Controversies arose at the beginning over whether the DES key was long enough to prevent the code's being broken by an enumerative search for the key, and whether the code contained secret trapdoors that would permit the government to read DES ciphers. Those controversies have quieted; no trapdoors have been found. Double or triple encryption with different keys can be used for extra protection. Because the DES is now ten years old, cryptographers have begun to seek

replacements suitable for commercial use.

Another kind of cryptosystem was proposed in 1976 by Whitfield Diffie and Martin Hellman of Stanford University. They called theirs a public-key cryptosystem to distinguish it from the traditional private-key systems. The public-key system uses two complementary keys: one is made public and is used to encipher messages; the other is kept secret and is used to decipher messages. The secret key cannot be deduced from the public key. Single-key cryptosystems are symmetric because the same key is used for both enciphering and deciphering; two-key cryptosystems are asymmetric. In a symmetric cryptosystem, almost any binary pattern can serve as a key, but a good deal of computation is required to generate a pair of keys for an asymmetric cryptosystem.

The notation for a public-key system is straightforward. A principal A holds secret and public keys, denoted SA and PA . To communicate with A , B sends the ciphertext $[M]^{PA}$; A recovers the message by deciphering the ciphertext with the secret key, because $M = [[M]^{PA}]^{SA}$. A and B can hold a conversation by exchanging messages enciphered under each other's public keys. Secrecy is assured because there is only one copy of the secret key, held by the principal who generated it.

Secrecy and authentication are separated in a two-key cryptosystem. Secrecy results from enciphering with the recipient's public key: anyone can generate $[M]^{PA}$, but only A can decipher it. Authentication results from enciphering with the sender's secret key: only A can generate $[M]^{SA}$, and anyone can

decipher it. Two encipherments are needed to provide both: $[[M]^{SA}]^{PB}$ can be enciphered only by A and deciphered only by B .

The first public-key cryptosystem with these properties was devised in 1977 by Ronald Rivest, Adi Shamir, and Len Adleman of MIT, and is known by their initials, RSA. It works as follows: To generate a key, pick two large prime numbers p and q . Then choose two integers d and e so that $de \bmod (p-1)(q-1) = 1$. (In general, $x \bmod y$ means the remainder after dividing x by y .) Let $n = pq$. The secret key is (d, n) and the public key is (e, n) . To encipher, compute $C = [M]^{PA} = M^e \bmod n$. To decipher, compute $M = [C]^{SA} = C^d \bmod n$. Deciphering recovers M because of a classical theorem of Fermat that says $M^{de} \bmod n = M$.

As an example, suppose $p = 3$ and $q = 11$; then $n = 33$ and $(p-1)(q-1) = 20$. Pick $(d, e) = (3, 7)$; this is valid because $de \bmod 20 = 21 \bmod 20 = 1$. Suppose $M = 4$; the ciphertext is then $C = 16$, because $4^7 \bmod 33 = 16384 \bmod 33 = (33 \times 496 + 16) \bmod 33 = 16$. The deciphered message is $M = 4$, because $16^3 \bmod 33 = 4096 \bmod 33 = (33 \times 124 + 4) \bmod 33 = 4$.

The security of the RSA system relies on the extreme difficulty of factoring a large composite number: If the prime components p and q could be recovered easily from n , a deciphering key matching the public enciphering key could be computed easily. In the summer of 1986, researchers at the Mitre Corporation factored an 84-digit number, the largest ever, after several days of computation on a set of cooperating computers. To protect against faster supercomputers

and improved factoring algorithms, most designers of RSA systems recommend that n be on the order of 200 digits (about 665 bits).

Computer chips containing the RSA algorithm have been developed. Because of the large number of digits in each block of enciphered data (around 200), these chips are rather slow, operating on the order of a few kilobits per second. This means that known public-key systems are too slow for high-bandwidth, secret conversations between computers.

What, then, is the advantage of a public-key system? It is the ability to separate authentication from secrecy. This separation permits digital signatures, which allow third parties to certify the identity of a sender. It works as follows: A signed message consists of a header H , a body M , and a signature block $X = [F(M)]^{SA}$; the header asserts that the message came from some sender, say A ; the signature is a small block computed from M and then signed with A 's secret key. The data-compression function F , often called a hashing function, is public; its result, $F(M)$, is called a checksum. The receiver will accept the message only if the signature, deciphered with A 's public key, is identical to the checksum of the message actually received. If A claims that B changed the message, or B claims that A sent a different message, a third party can resolve the dispute by deciphering the signature and comparing it to the claimed message's checksum. If the message is a secret, the message body can be the ciphertext $[M]^K$ and the enciphered key $[K]^{PB}$ can be added to the signature block.

The same principles work in broader arenas. Suppose the space station contains a telescope that emits a stream of data, which, by treaty, is supposed to be available to every astronomer in the world. How can an astronomer be assured that a data stream is in fact the one transmitted by the telescope, and that none of the data have been altered? The raw data can be collected in a local buffer in the space telescope, which is assigned a public key PT and secret key ST . Each buffer is treated as a message M ; when the buffer is full, the authenticator $[F(M)]^{ST}$ is appended, and the result is transmitted publicly. Any receiver can reverse the process and check that each block of data is authentic.

In 1978, Gus Simmons of Sandia Laboratories proposed a similar scheme for the verification of compliance with test-ban treaties. He assumed that the United States would require assurances that its monitoring device implanted in Soviet soil had not been tampered with, and the Soviets would want to be able to read the transmissions of the device.

There are many practical considerations to building secure signature systems that will work in large networks. For example, the hashing function must deprive potential intruders of effective means to construct fake messages with the same checksums as authentic messages. The subject is covered well in articles by Donald Davies and Dorothy Denning

A cryptosystem is useless unless distribution of keys is secure. Let us examine this problem for networks in which all conversations are protected by private-key cryptosystems. How are keys handed out so that the communicants

are sure of one another's identities? An obvious solution relies on a registry service R . A private key is generated for each principal A , one copy of which is stored in R and another copy on a key card (or other medium) that can be inserted into an encryption device attached to A . Now it is possible for R to provide A with private keys for conversations with other principals in the network. Roger Needham and Michael Schroeder have proposed protocols that allow any A and B , with help from R , to obtain a private key for a secure communication between them. Victor Vaydock and Stephen Kent have shown how to apply these protocols in real networks.

The dependability of networks is sensitive to the correct, reliable operation of key registries. The whole approach becomes unwieldy in large networks: Failures of registries can prevent principals from initiating new conversations and can compromise keys. Trust itself is a serious issue in a large network; the US and Soviet governments, for example, are not likely to believe that each other's registries will refrain from listening in on conversations for which they have passed out the keys.

The amount of faith required can be reduced by using public-key cryptography to exchange the private keys for conversations. Now the registry service becomes simply a directory service D . Principals can register public keys with D for later redistribution, but they do not need to reveal their secret keys to D . To converse with B , A consults D to obtain the public key P_B , generates a conversation key K , and sends $[K]^{P_B}$ to B with a request to open a conversa-

tion. A must also authenticate itself to B , which can be done with a certificate as discussed below. Now the responsibility for generating keys rests with the communicants, and the directory service has no special knowledge that would enable it to listen in on any conversations.

There is still a catch -- trusting the authenticity of public keys dispensed by the directory service or by any other principal. The authenticity of this information can be guaranteed by storing it as public-key certificates created, on request, by a network notary service. Certificates are messages of the form $[B, PB, T]^{SN}$, where SN is the secret key of the notary service and T is the time of the certificate's creation. Anyone can decipher a certificate using the notary's public key, thereby obtaining the public key of the principal identified therein. If for some reason the notary's secret key is compromised, all subsequently issued certificates are invalid. A good deal of effort must be put into protecting the notary's secret key, but the effort is worthwhile because the security of network communications does not rest on the trustworthiness of the directory service

The safety of remotely controlled experiments and integrity of research rest critically on the ability of the network to authenticate senders and receivers, to protect proprietary communications, and to sign some transmissions. Mathematically sound schemes for encrypting data and distributing keys make security an attainable goal.

References

1. Dorothy E. Denning. 1982. *Cryptography and Data Security*. Addison-Wesley. (Especially Chapters 1 and 2.)
2. Ronald Rivest, Adi Shamir, and Len Adleman. 1978. "A method for obtaining digital signatures and public-key cryptosystems". *ACM Communications*, Vol. 21, February.
3. Donald W. Davies. 1983. "Applying the RSA digital signature to electronic mail." *IEEE Computer*. February.
4. Dorothy E. Denning. 1983. "Protecting public keys and signature keys." *IEEE Computer*. February.
5. Roger M. Needham and Michael D. Schroeder. 1978. "Using encryption for authentication in large networks of computers." *ACM Communications*, Vol. 21, December.
6. Victor L. Voydock and Stephen T. Kent. 1983. "Security mechanisms in high-level network protocols." *ACM Computing Surveys*, Vol 15, June.